

Politechnika Świętokrzyska  
Wydział Elektrotechniki, Automatyki i Informatyki

Zespół:  
**Kot Jarosław**  
**Prusicki Jakub**

## **Łamacz haseł**

Projekt zespołowy  
na studiach stacjonarnych  
o kierunku **Informatyka**  
Stopień II

Opiekun projektu:  
**dr inż. Paweł Paduch**

Kielce, 2023

## 1. NOWY ALGORYTM BRUTE FORCE.

Aby spełnić założenia projektowe niezbędnym było odpowiednie zaprojektowanie algorytmu BruteForce, który potrafiłby łamać kolejne hasła. Program powinien działać na podstawie porównywania kolejnych haseł (otrzymanych np. w postaci listy) od serwera, a następnie porównywać je z wszystkimi możliwymi permutacjami wygenerowanymi na podstawie zadanego alfabetu oraz zakresów.

```
using System;
using System.Collections.Generic;

namespace BruteForceApp
{
    1 reference
    class BruteForce
    {
        1 reference
        public static void Run(List<string> passwords, string alphabet, string startRange, string endRange)
        {
            List<string> listOfPermutations = PermutationsGenerator.GeneratePermutationsWithRepetitions(alphabet, startRange, endRange); //generowanie permutacji

            passwords = MD5.CreateListOfMD5(passwords); //pobiera w parametrze listę haseł, zamienia ją na listę hash'y i zapisuje do tej samej listy
            listOfPermutations = MD5.CreateListOfMD5(listOfPermutations); //jak powyżej ... tylko dla listy permutacji

            ComparePasswordsListWithHashes(passwords, listOfPermutations);
        }

        private static void ComparePasswordsListWithHashes(List<string> passwords, List<string> listOfPermutations)
        {
            foreach (string password in passwords)
            {
                if (CompareHash(password, listOfPermutations))
                {
                    // ----- !!! HASŁO ZNALEZIONE !!! -----
                    Console.WriteLine("Hasło złamane");
                }
            }
        }

        1 reference
        private static bool CompareHash(string password, List<string> listOfPermutations)
        {
            foreach (string permutation in listOfPermutations)
            {
                if (password.Equals(permutation))
                {
                    return true;
                }
            }

            return false;
        }
    }
}
```

Rys. 1. Nowa klasa BruteForce.

Warto podkreślić, że poprzednie wersje generowały możliwe permutacje na podstawie całego alfabetu, a nie tak jak teraz tylko dzięki odpowiednim zakresom. Pozwala to na odciążenie komputerów z wykonywania zbędnej pracy oraz rozłożenie jej na pozostałe komputery. Dzięki temu program staje się bardziej elastyczny i działa na lepszym poziomie optymalizacji. Jeśli chodzi o postępy w tworzonej oprogramowaniu, to nie można pominąć faktu iż program został wdrożony do wersji z rozproszeniem sieciowym (Web-API).